Machine Learning Kernels in Julia

trthatcher

May 3, 2015

Let $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ be a kernel function. There are several classes that can be identified. This focusses on kernels of the form:

$$k(\mathbf{x}, \mathbf{y}) = \kappa(z(\mathbf{x}, \mathbf{y}))$$

where $z : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ and $\kappa : \mathbb{R} \mapsto \mathbb{R}$.

For the first derivatives:

$$\frac{\partial \kappa}{\partial \mathbf{x}} = \kappa'(z) \left[\frac{\partial z}{\partial \mathbf{x}}\right] \quad \text{and} \quad \frac{\partial \kappa}{\partial \mathbf{y}} = \kappa'(z) \left[\frac{\partial z}{\partial \mathbf{y}}\right]$$

Then, the mixed second derivatives are:

$$\begin{aligned}
\frac{\partial \kappa^2}{\partial \mathbf{x} \partial \mathbf{y}} &= \frac{\partial}{\partial \mathbf{x}} \left(\left[\frac{\partial \kappa}{\partial \mathbf{y}}\right] \right) \\
&= \frac{\partial}{\partial \mathbf{x}} \left(\kappa'(z) \left[\frac{\partial z}{\partial \mathbf{y}}\right] \right) \\
&= \left[\frac{\partial z}{\partial \mathbf{y}}\right] \left[\frac{\partial \kappa'(z)}{\partial \mathbf{x}}\right]^{\intercal} + \kappa'(z) \frac{\partial}{\partial \mathbf{x}} \left(\left[\frac{\partial z}{\partial \mathbf{y}}\right] \right) \\
&= \kappa''(z) \left[\frac{\partial z}{\partial \mathbf{y}}\right] \left[\frac{\partial z}{\partial \mathbf{x}}\right]^{\intercal} + \kappa'(z) \left[\frac{\partial z^2}{\partial \mathbf{x} \partial \mathbf{y}}\right]
\end{aligned}$$

1 Scalar Product Kernel: $z(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\intercal}\mathbf{y}$

The Scalar Product kernel is of the form:

$$k(x, y) = \kappa\left(x^{\intercal} y\right)$$

where κ is a function unique to each Scalar Product kernel. For example, the the Sigmoid kernel instance of κ is defined to be:

$$\kappa_{\mathrm{s}}(z; a, c) = \tanh(az + c)$$

Let X and Y be data matrices with rows as data points. Then the kernel matrix K:

$$X = \begin{bmatrix} \mathbf{x}_1^\intercal \\ \mathbf{x}_2^\intercal \\ \vdots \\ \mathbf{x}_n^\intercal \end{bmatrix} \text{ and } Y = \begin{bmatrix} \mathbf{y}_1^\intercal \\ \mathbf{y}_2^\intercal \\ \vdots \\ \mathbf{y}_m^\intercal \end{bmatrix} \quad \Longrightarrow \quad K = \begin{bmatrix} \kappa(\mathbf{x}_1^\intercal \mathbf{y}_1) & \cdots & \kappa(\mathbf{x}_1^\intercal \mathbf{y}_m) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_n^\intercal \mathbf{y}_1) & \cdots & \kappa(\mathbf{x}_n^\intercal \mathbf{y}_m) \end{bmatrix}$$

Given a set of vectors $S = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ the gramian is defined as $[\mathbf{G}]_{ij} = \mathbf{z}_i^\intercal \mathbf{z}_j$. Therefore, if we have two sets of vectors S_x and S_y with corresponding data matrices X and Y, then we can define the gramian between them as $\mathbf{G}(\mathbf{X}, \mathbf{Y}) = \mathbf{X}\mathbf{Y}^\intercal$. This is really the upper right-hand corner of the gramian of the set $S_x \cup S_y$.

Finally, we may define the kernel matrix as a function of the gramian:

$$[K]_{ij} = \kappa\left([G_{XY}]_{ij}\right)$$

Generally, the best approach to take for implementation is:

1. Calculate G_{XY} for matrix X and Y using GEMM or SYRK using BLAS (exploit symmetry of kernels). The option to use rows or columns as data points is left as an option (using trans = 'N' assumes rows are observations and 'T' assumes columns). This is $O(n^3)$ but is heavily optmized to reduce the coefficient.
2. Scan through the elements of $\mathbf{G}(\mathbf{X}, \mathbf{Y})$ and apply the κ function to each element, overwrite G_{XY} in the process. This is $O(n^2)$.
3. Return K

2 Squared Distance Kernel

The Squared Distance kernel is of the form:

$$k(x, y) = \kappa\left((||x - y||^2\right)$$

where κ is a function unique to each kernel. For example, the the Gaussian kernel instance of κ is defined to be:

$$\kappa_\Phi(z; \sigma) = \exp\left(\frac{-z}{2\sigma^2}\right)$$

2.1 Kernel Matrix

Let $\mathbf{X}$ and $\mathbf{Y}$ be data matrices with rows as data points. Then the kernel matrix K:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\intercal \\ \mathbf{x}_2^\intercal \\ \vdots \\ \mathbf{x}_n^\intercal \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^\intercal \\ \mathbf{y}_2^\intercal \\ \vdots \\ \mathbf{y}_m^\intercal \end{bmatrix} \quad \Longrightarrow \quad \mathbf{K} = \begin{bmatrix} \kappa\left(||\mathbf{x}_1 - \mathbf{y}_1||^2\right) & \cdots & \kappa\left(||\mathbf{x}_1 - \mathbf{y}_m||^2\right) \\ \vdots & \ddots & \vdots \\ \kappa\left(||\mathbf{x}_n - \mathbf{y}_1||^2\right) & \cdots & \kappa\left(||\mathbf{x}_n - \mathbf{y}_m||^2\right) \end{bmatrix}$$

For a vector $\mathbf{x}$ and $\mathbf{y}$, define the lag (alternatively, the error) between the two vectors to be:

$$\text{lag}(x, y) = x - y = \epsilon$$

Then we may define kernels of this form as:

$$k(x, y) = \kappa \left(\text{lag}(\mathbf{x}, \mathbf{y}) \cdot \text{lag}(\mathbf{x}, \mathbf{y})\right) = \kappa \left(\epsilon^\intercal \epsilon\right) = \kappa \left(||\mathbf{x} - \mathbf{y}||^2\right)$$

Given a set of vectors $S = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$, the lagged gramian is defined as:

$$[\mathbf{G}_l]_{ij} = ||\mathbf{z}_i - \mathbf{z}_j||^2$$

Similarly, for two data matrices $\mathbf{X}$ and $\mathbf{Y}$:

$$[\mathbf{G}_l \left(\mathbf{X}, \mathbf{Y}\right)]_{ij} = ||\mathbf{x}_i - \mathbf{y}_j||^2$$

Where $\mathbf{x}_i$ and $\mathbf{y}_j$ are the data vectors in $\mathbf{X}$ and $\mathbf{Y}$. The lagged gramian can be defined in terms of $\mathbf{G}(\mathbf{X}, \mathbf{Y})$, $\mathbf{X}$ and $\mathbf{Y}$.

$$[\mathbf{G}_l \left(\mathbf{X}, \mathbf{Y}\right)]_{ij} = [\text{diag} \left(\mathbf{G}(\mathbf{X})\right)]_i - 2 \left[\mathbf{G}\right]_{ij} + [\text{diag} \left(\mathbf{G}(\mathbf{Y})\right)]_j$$

The function $\text{diag}\left(\mathbf{G}(\mathbf{X})\right)$ returns a vector of diagonal entries of $\mathbf{G}$. This is simply a vector where entry i is $\mathbf{x}_i^\intercal \mathbf{x}_i$.

2.2 Kernel Matrix Calculation - Implementation

Therefore, to compute the kernel matrix for a squared distance kernel:

1. Calculate $\mathbf{G}(\mathbf{X}, \mathbf{Y})$ for matrix $\mathbf{X}$ and $\mathbf{Y}$ using GEMM or SYRK using BLAS (exploit symmetry of kernels). This is $O(n^3)$.
2. Transform $\mathbf{G}(\mathbf{X}, \mathbf{Y})$ to $\mathbf{G}_l(\mathbf{X}, \mathbf{Y})$. This is $O(n^2)$.
3. Scan through the elements of $\mathbf{G}_l(\mathbf{X}, \mathbf{Y})$ and apply the κ function to each element, overwrite $\mathbf{G}_l(\mathbf{X}, \mathbf{Y})$ in the process. This is $O(n^2)$.
4. Return K

2.3 Kernel Derivative

$$\frac{\partial z}{\partial \mathbf{x}} = \frac{\partial ||\mathbf{x} - \mathbf{y}||^2}{\partial \mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{x}^\intercal \mathbf{x} - 2\mathbf{x}^\intercal \mathbf{y} + \mathbf{y}^\intercal \mathbf{y}\right) = 2(\mathbf{x} - \mathbf{y})$$
$$\frac{\partial z}{\partial \mathbf{y}} = \frac{\partial ||\mathbf{x} - \mathbf{y}||^2}{\partial \mathbf{y}} = \frac{\partial}{\partial \mathbf{y}} \left(\mathbf{x}^\intercal \mathbf{x} - 2\mathbf{x}^\intercal \mathbf{y} + \mathbf{y}^\intercal \mathbf{y}\right) = 2(\mathbf{y} - \mathbf{x})$$

Using the identity:

$$\left[\frac{\partial \mathbf{u}}{\partial \mathbf{v}}\right]_{ij} = \frac{\partial u_i}{\partial v_i}$$

The second mixed derivative is:

$$\frac{\partial z^2}{\partial \mathbf{x} \partial \mathbf{y}} = \frac{\partial ||\mathbf{x} - \mathbf{y}||^2}{\partial \mathbf{x} \partial \mathbf{y}} = \frac{\partial}{\partial \mathbf{x}} 2(\mathbf{y} - \mathbf{x}) = -2\mathbf{I}_d$$

Substituting in the above into the formula for kernel derivative:

$$\begin{aligned}\frac{\partial \kappa^2}{\partial \mathbf{x} \partial \mathbf{y}} &= \kappa''(z)\left[\frac{\partial z}{\partial \mathbf{y}}\right]\left[\frac{\partial z}{\partial \mathbf{x}}\right]^{\intercal} + \kappa'(z)\left[\frac{\partial z^2}{\partial \mathbf{x} \partial \mathbf{y}}\right] \\ &= -4\kappa''(z)(\mathbf{x}-\mathbf{y})(\mathbf{x}-\mathbf{y})^{\intercal} - 2\kappa'(z)\mathbf{I}_d\end{aligned}$$

Define $\boldsymbol{\epsilon} = \mathbf{x} - \mathbf{y}$, then this gives an element-wise formula of:

$$\left[\frac{\partial \kappa^2}{\partial \mathbf{x} \partial \mathbf{y}}\right]_{ij} = \begin{cases} -4\kappa''(\boldsymbol{\epsilon}^{\intercal}\boldsymbol{\epsilon})\epsilon_i\epsilon_j & \text{if } i \neq j \\ -4\kappa''(\boldsymbol{\epsilon}^{\intercal}\boldsymbol{\epsilon})\epsilon_i^2 - 2\kappa'(\boldsymbol{\epsilon}^{\intercal}\boldsymbol{\epsilon}) & \text{if } i = j \end{cases}$$

2.4 Kernel Derivative - Implementation

To compute $\left[\frac{\partial \kappa^2}{\partial \mathbf{x} \partial \mathbf{y}}\right]_{ij}$:

1. Compute array $\mathbf{A}$ of element-wise differences for all coordinates in arrays $\mathbf{X}$ and $\mathbf{Y}$.
2. Compute the lagged gramian matrix
3. Scan through the elements of $\mathbf{A}$ and apply the κ functions
4. Return K